

Winning Space Race with Data Science

Kotha Shiva Krishna Reddy
12th February 2024



Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- **Summary of methodologies**

- Data Collection through API
- Data Collection with Web Scrapping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

- **Summary of all results**

- Exploratory Data Analysis result
- Interactive analytics
- Predictive Analysis

Introduction

- Project background and context

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

- What factors determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
 - Data was collected using SpaceX API and web scrapping from Wikipedia
- Perform data wrangling
 - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - How to build, tune, evaluate classification models

Data Collection

- The data was collected using various methods
 - Data collection was done using get request to the SpaceX API.
 - Next, we decoded the response content as a Json using `.json()` function call and turn it into a pandas dataframe using `.json_normalize()`.
 - We then cleaned the data, checked for missing values and fill in missing values where necessary.
 - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
 - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- Used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- <https://github.com/ShivaKrishnaRddyKotha/DataScienceCapstone/blob/main/spacex-data-collection-api.ipynb>

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
In [9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/SpacexAPI.json'
```

We should see that the request was successful with the 200 status response code

```
In [10]: response.status_code
```

```
Out[10]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
In [11]: # Use json_normalize method to convert the json result into a dataframe  
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
In [12]: # Get the head of the dataframe  
data.head()
```

```
Out[12]: static_fire_date_utc static_fire_date_unix net window rocket success failures details crew  
0 2006-03-17T00:00:00.000Z 1.142554e+09 False 0.0 5e9d0d95eda69955f709d1eb False 'reason': [{"time": 33, "altitude": None, "reason": "Engine failure at 33 seconds"}]
```

Data Collection - Scraping

- Applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- Parsed the table and converted it into a pandas dataframe.
- <https://github.com/ShivaKrishnaReddyKotha/DataScienceCapstone/blob/main/webscraping.ipynb>

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
✓ 0s

Next, request the HTML page from the above URL and get a response object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
✓ 7.2s

Create a BeautifulSoup object from the HTML response

# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
✓ 0.3s

Print the page title to verify if the BeautifulSoup object was created properly

# Use soup.title attribute
print(soup.title)
✓ 0.0s

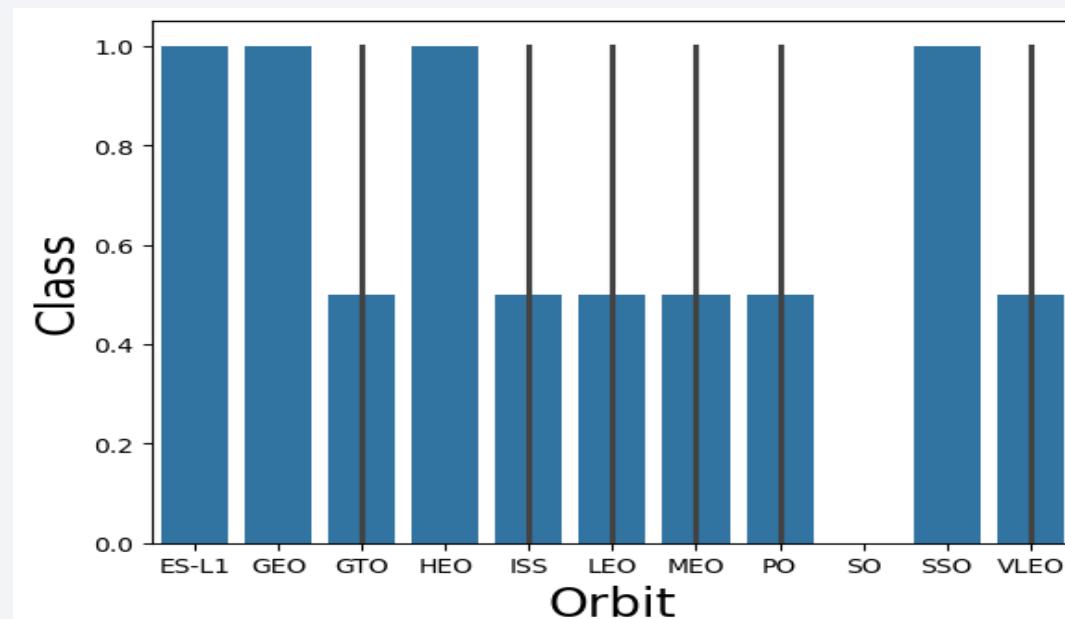
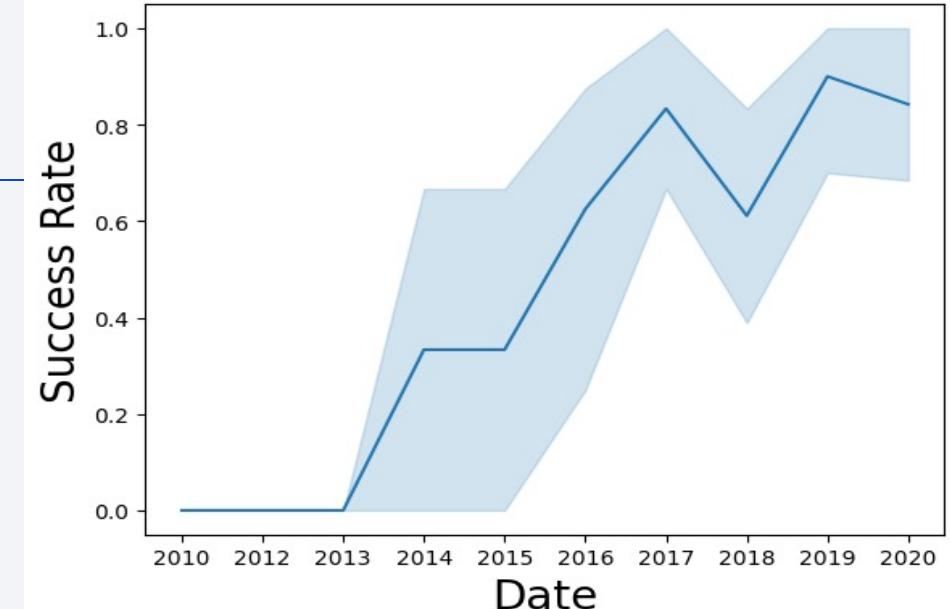
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

- Performed exploratory data analysis and determined the training labels.
- Calculated the number of launches at each site, and the number and occurrence of each orbits
- Created landing outcome label from outcome column and exported the results to csv.
- <https://github.com/ShivaKrishnaReddyKotha/DataScienceCapstone/blob/main/spacex-Data%20wrangling.ipynb>

EDA with Data Visualization

- Explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- <https://github.com/ShivaKrishnaReddyKotha/DataScienceCapstone/blob/main/ExploratoryDataAnalysis-datavizualization.ipynb>



EDA with SQL

- Loaded the SpaceX dataset into a SQL database without leaving the jupyter notebook.
- Applied EDA with SQL to get insight from the data and wrote queries to find out for instance:
 - The names of unique launch sites in the space mission.
 - The total payload mass carried by boosters launched by NASA (CRS)
 - The average payload mass carried by booster version F9 v1.1
 - The total number of successful and failure mission outcomes
 - The failed landing outcomes in drone ship, their booster version and launch site names.
- <https://github.com/ShivaKrishnaReddyKotha/DataScienceCapstone/blob/main/ExploratoryDataAnalysis-sql.ipynb>

Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- Calculated the distances between a launch site to its proximities and answered some question for instance:
 - Are launch sites near railways, highways and coastlines.
 - Do launch sites keep certain distance away from cities.

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash
- Plotted pie charts showing the total launches by a certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- [https://github.com/ShivaKrishnaReddyKotha/DataScienceCapstone/blob/main/
spacex_dash_app.py](https://github.com/ShivaKrishnaReddyKotha/DataScienceCapstone/blob/main/spacex_dash_app.py)

Predictive Analysis (Classification)

- Loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- Found the best performing classification model.
- https://github.com/ShivaKrishnaReddyKotha/DataScienceCapstone/blob/main/SpaceX_Machine_Learning_Prediction.ipynb

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

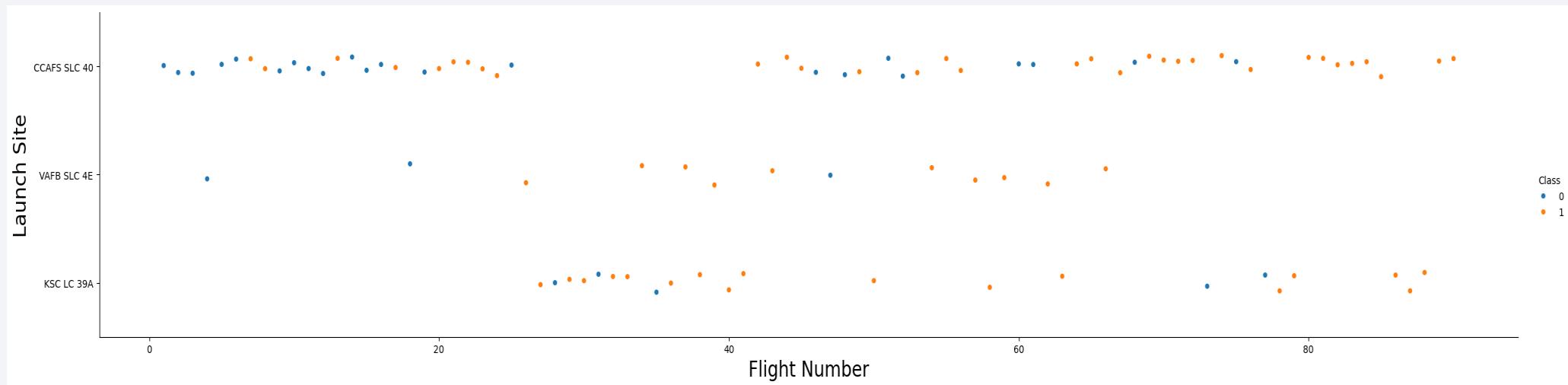
The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a three-dimensional space or a network of data points. The overall effect is futuristic and dynamic.

Section 2

Insights drawn from EDA

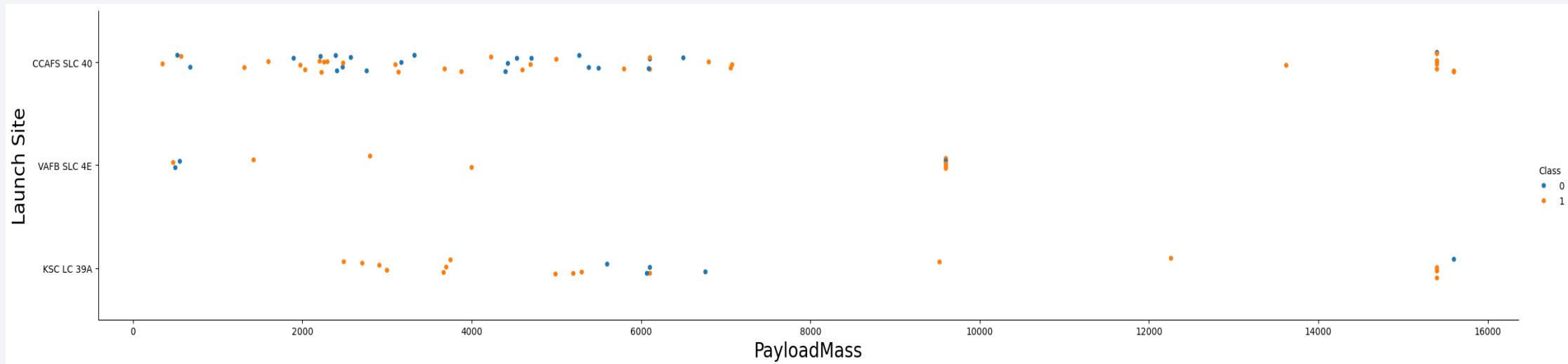
Flight Number vs. Launch Site

- From the plot, found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



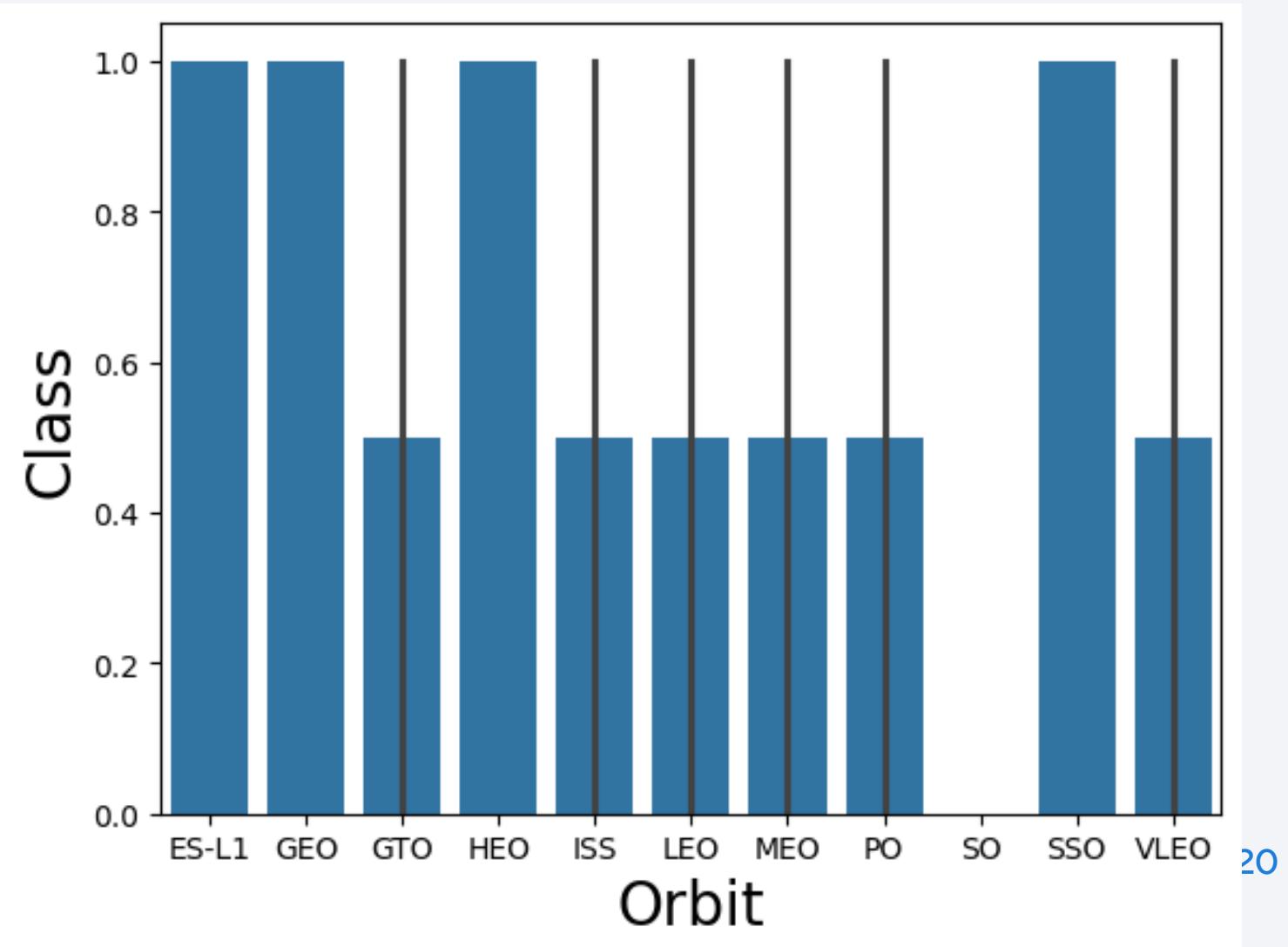
Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



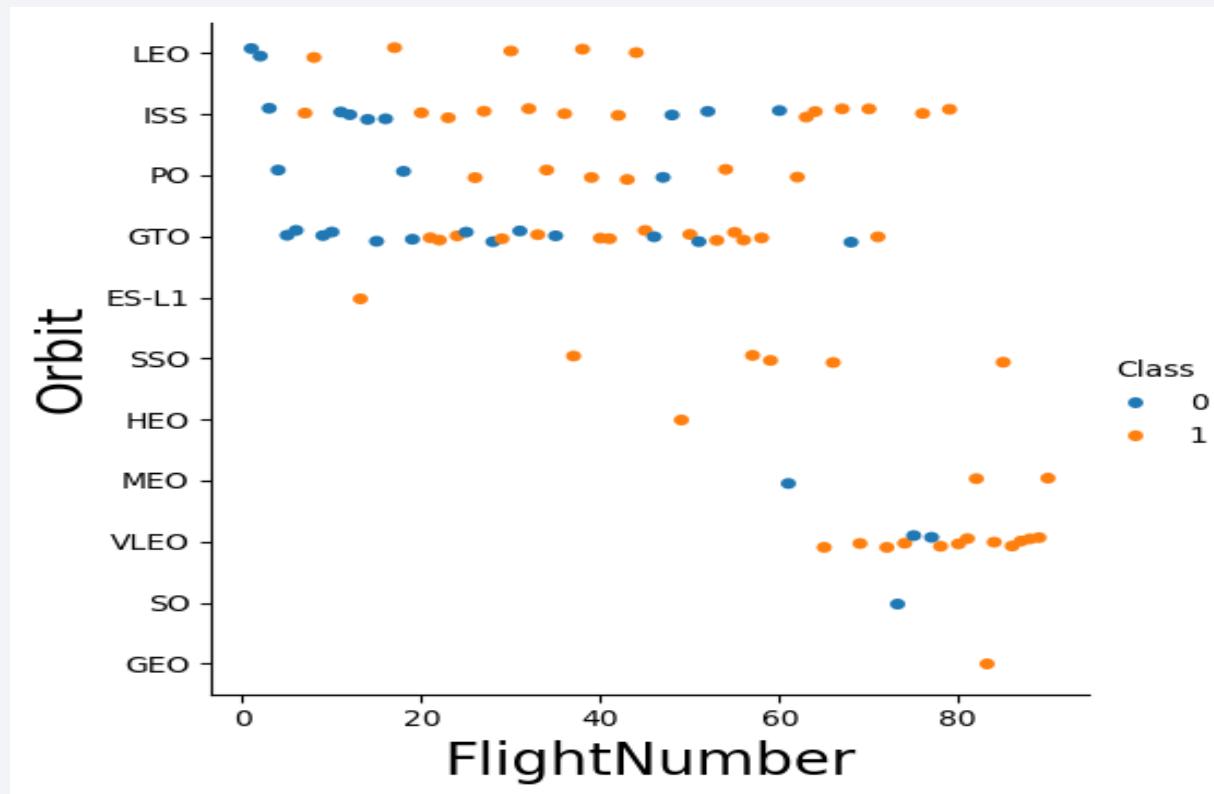
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had most success rate



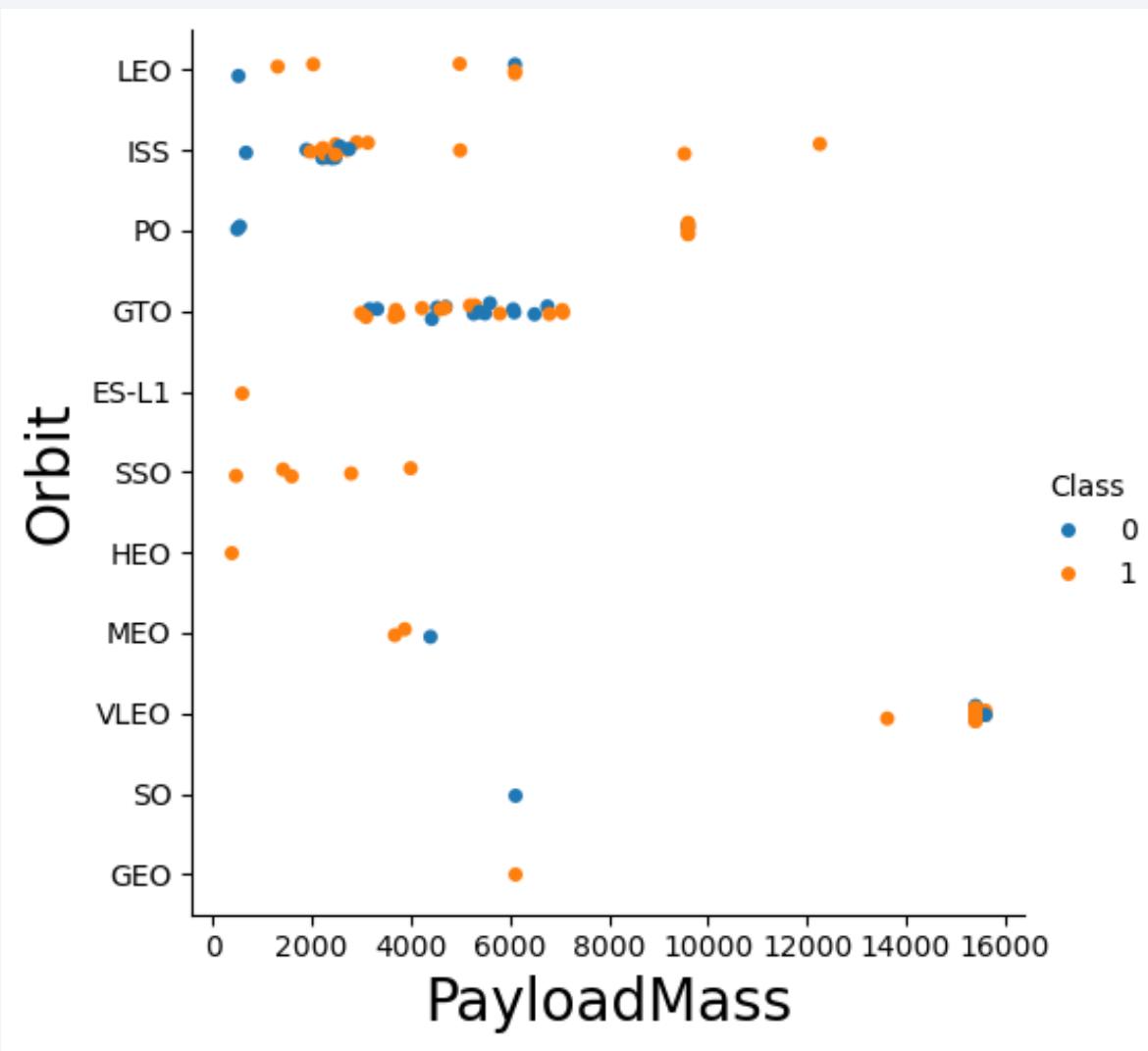
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



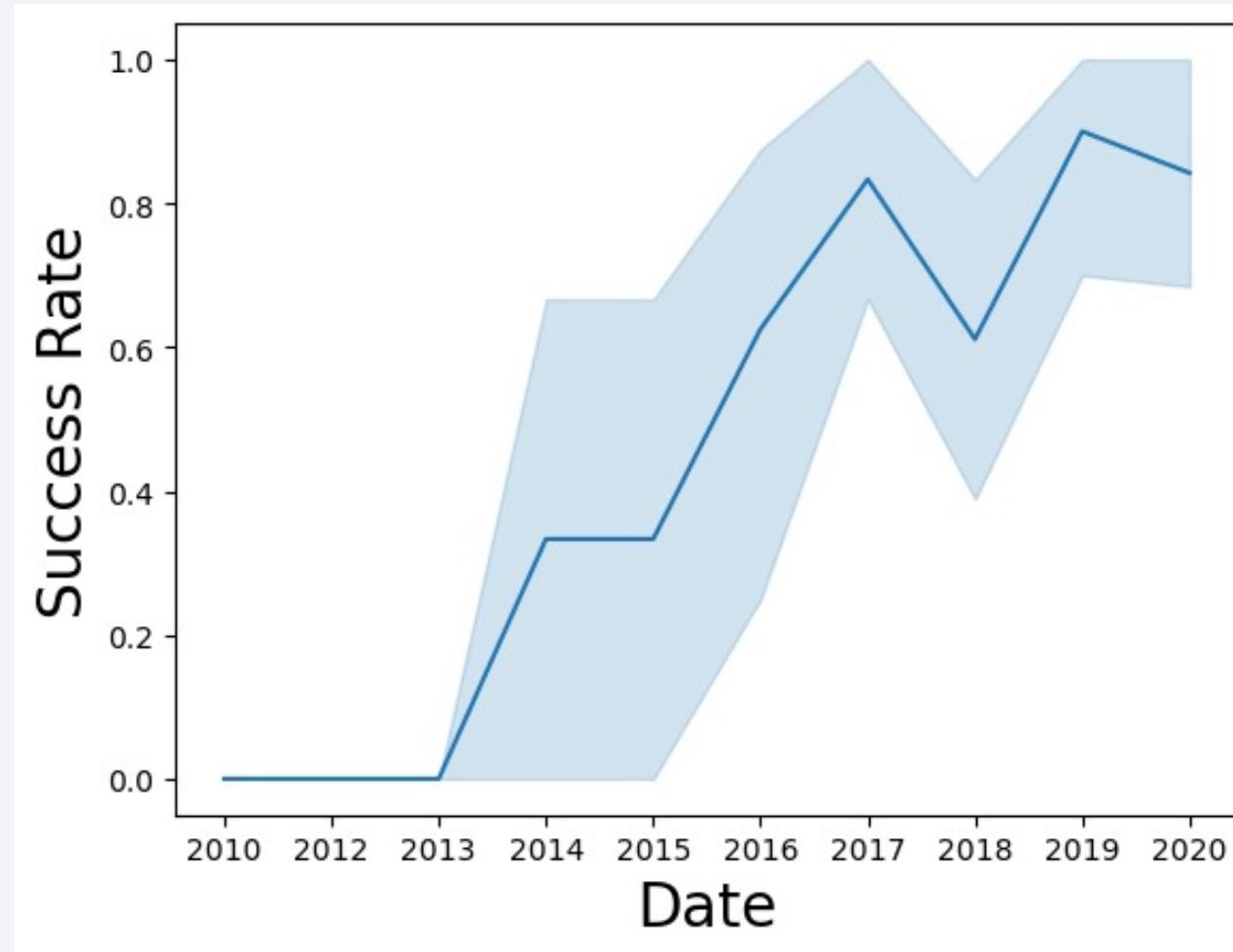
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- Used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

Task 1

Display the names of the unique launch sites in the space mission

```
In [8]: %sql select distinct(LAUNCH_SITE) from SPACEXTBL  
* sqlite:///my_data1.db  
Done.
```

```
Out[8]: Launch_Site
```

```
CCAFS LC-40
```

```
VAFB SLC-4E
```

```
KSC LC-39A
```

```
CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

In [9]:

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

* sqlite:///my_data1.db

Done.

Out[9]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (¶)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (¶)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	¶
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	¶
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	¶

Total Payload Mass

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

In [10]:

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

```
* sqlite:///my_data1.db  
Done.
```

Out[10]: sum(PAYLOAD_MASS__KG_)

45596

Average Payload Mass by F9 v1.1

Task 4

Display average payload mass carried by booster version F9 v1.1

```
In [11]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like 'F9 v1.1'  
* sqlite:///my_data1.db  
Done.
```

```
Out[11]: avg(PAYLOAD_MASS__KG_)  
2928.4
```

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
In [13]: %sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

```
* sqlite:///my_data1.db  
Done.
```

```
Out[13]: min(DATE)
```

```
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
In [17]: %sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ >
```

```
* sqlite:///my_data1.db  
Done.
```

```
Out[17]: Booster_Version
```

```
F9 FT B1022
```

```
F9 FT B1026
```

```
F9 FT B1021.2
```

```
F9 FT B1031.2
```

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [18]:

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure'
```

```
* sqlite:///my_data1.db  
Done.
```

Out[18]: count(MISSION_OUTCOME)

99

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [19]:

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

* sqlite:///my_data1.db
Done.

Out[19]: **Booster_Version**

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

```
In [30]: %%sql select distinct(Mission_Outcome) from SPACEXTBL
```

```
* sqlite:///my_data1.db  
Done.
```

```
Out[30]: Mission_Outcome
```

```
Success
```

```
Failure (in flight)
```

```
Success (payload status unclear)
```

```
Success
```

```
In [31]:
```

```
%%sql  
SELECT  
    substr(Date, 6, 2) AS Month,  
    Mission_Outcome,  
    Landing_Outcome,  
    Booster_Version,  
    Launch_Site  
FROM  
    SPACEXTBL  
WHERE  
    substr(Date, 0, 5) = '2015'  
    AND Mission_Outcome = 'Failure (in flight)'  
    AND Landing_Outcome = 'Failure (drone ship)';
```

```
* sqlite:///my_data1.db  
Done.
```

```
Out[31]: Month Mission_Outcome Landing_Outcome Booster_Version Launch_Site
```

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.
- Applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

In [33]:

```
%sql select * from SPACEXTBL where Landing_Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20')
```

* sqlite:///my_data1.db
Done.

Out [33]:

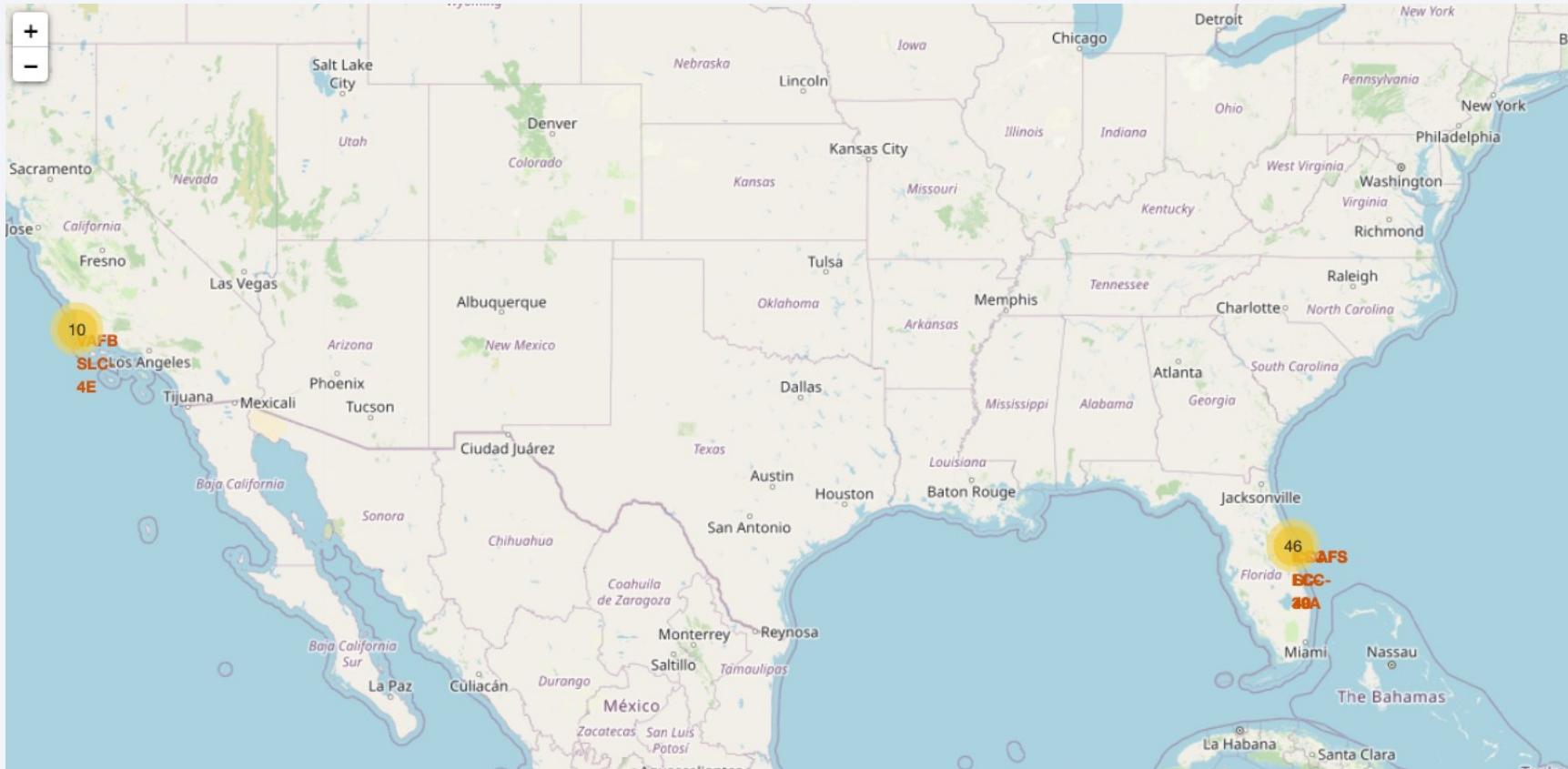
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_O
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Si
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Si
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Si
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Si
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Si
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Si
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Si
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 211 Orbcomm OG2	2034	LEO	Orbcomm	Success	Si

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth against a dark blue-black void of space. City lights are visible as numerous small white and yellow dots, primarily concentrated in the lower right quadrant where the United States appears. In the upper right, the green and yellow glow of the aurora borealis is visible. The atmosphere of the Earth is thin and hazy, appearing as a light blue band near the horizon.

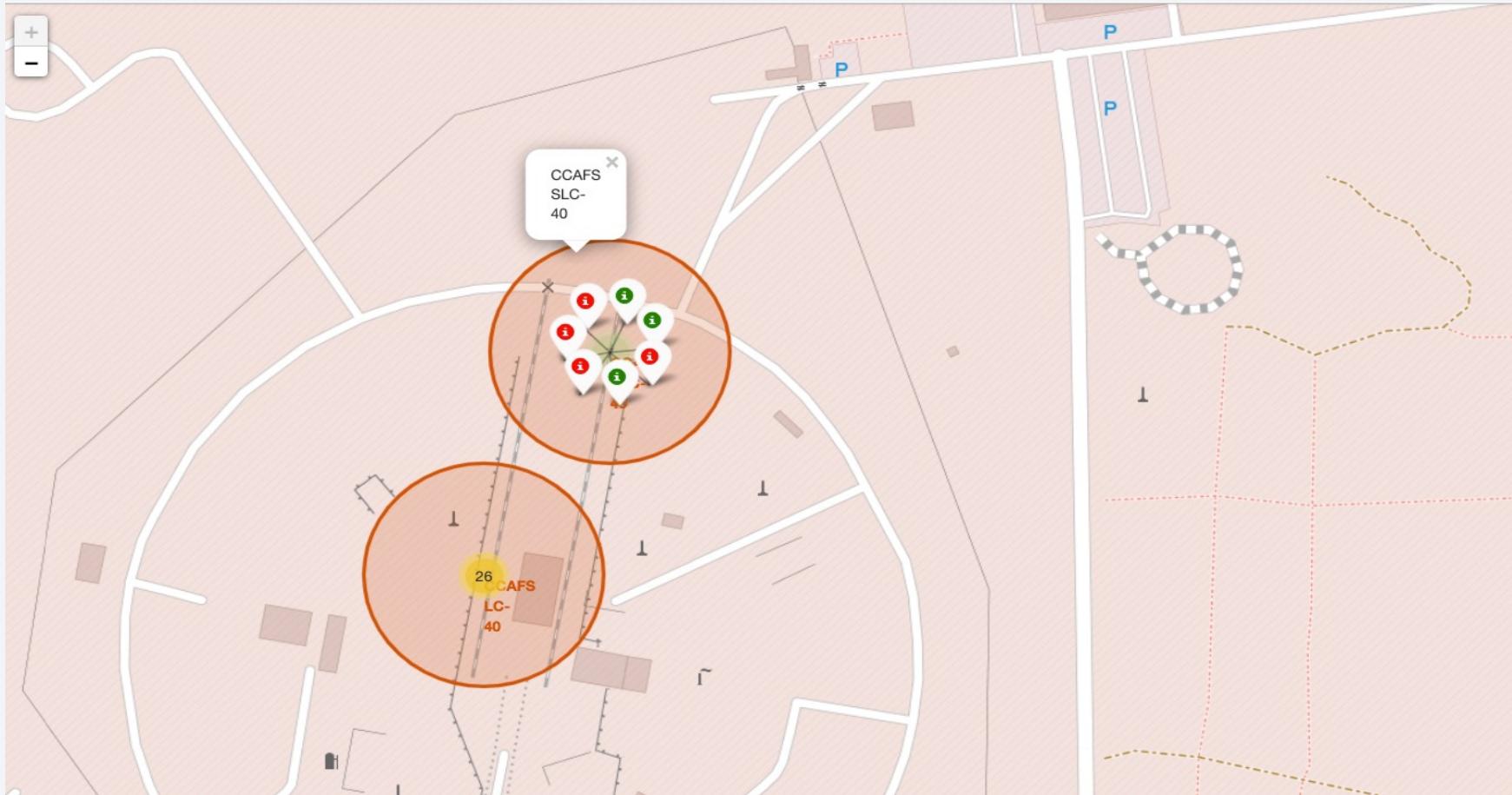
Section 3

Launch Sites Proximities Analysis

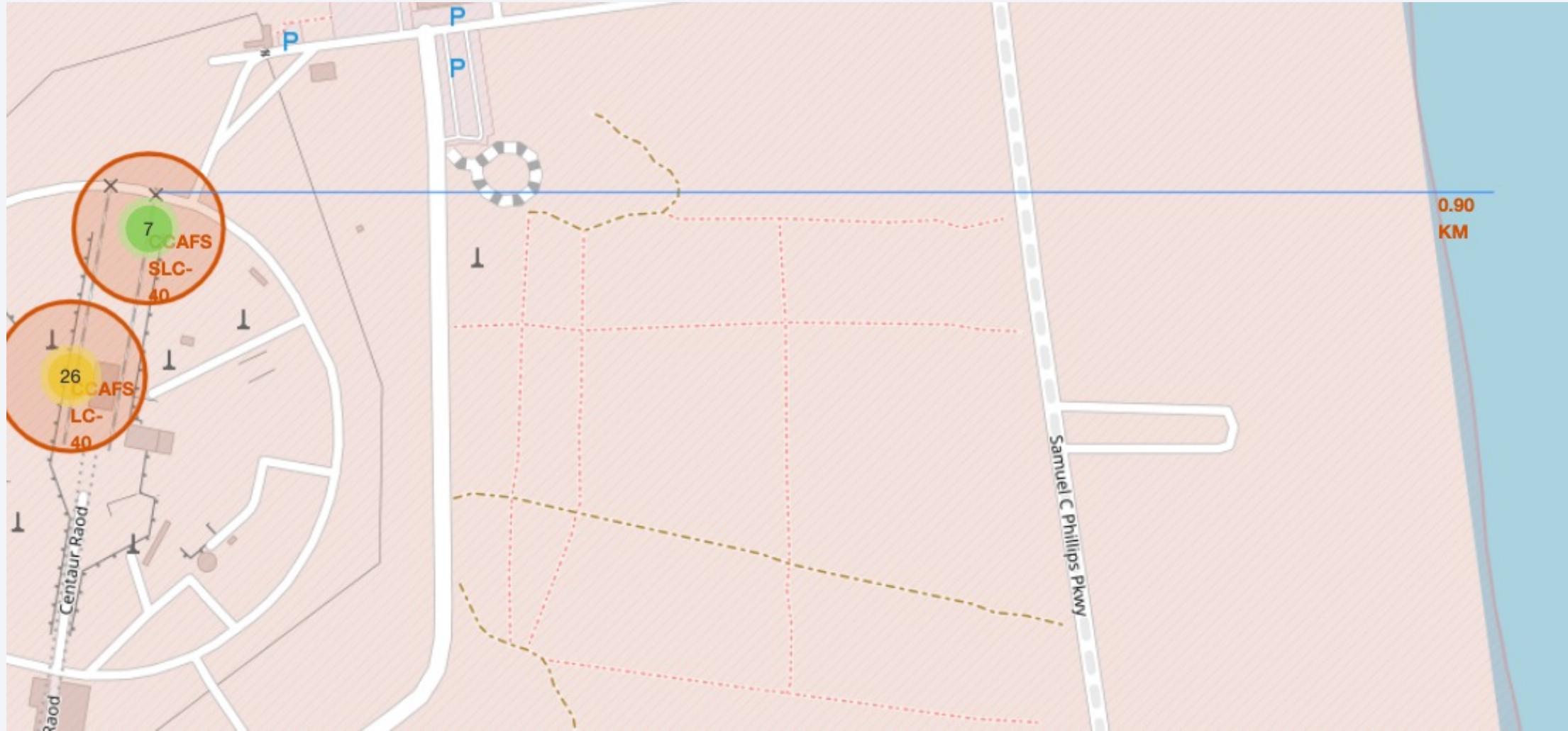
All launch sites global map markers



Markers showing launch sites with color labels



Launch Site distance to landmarks

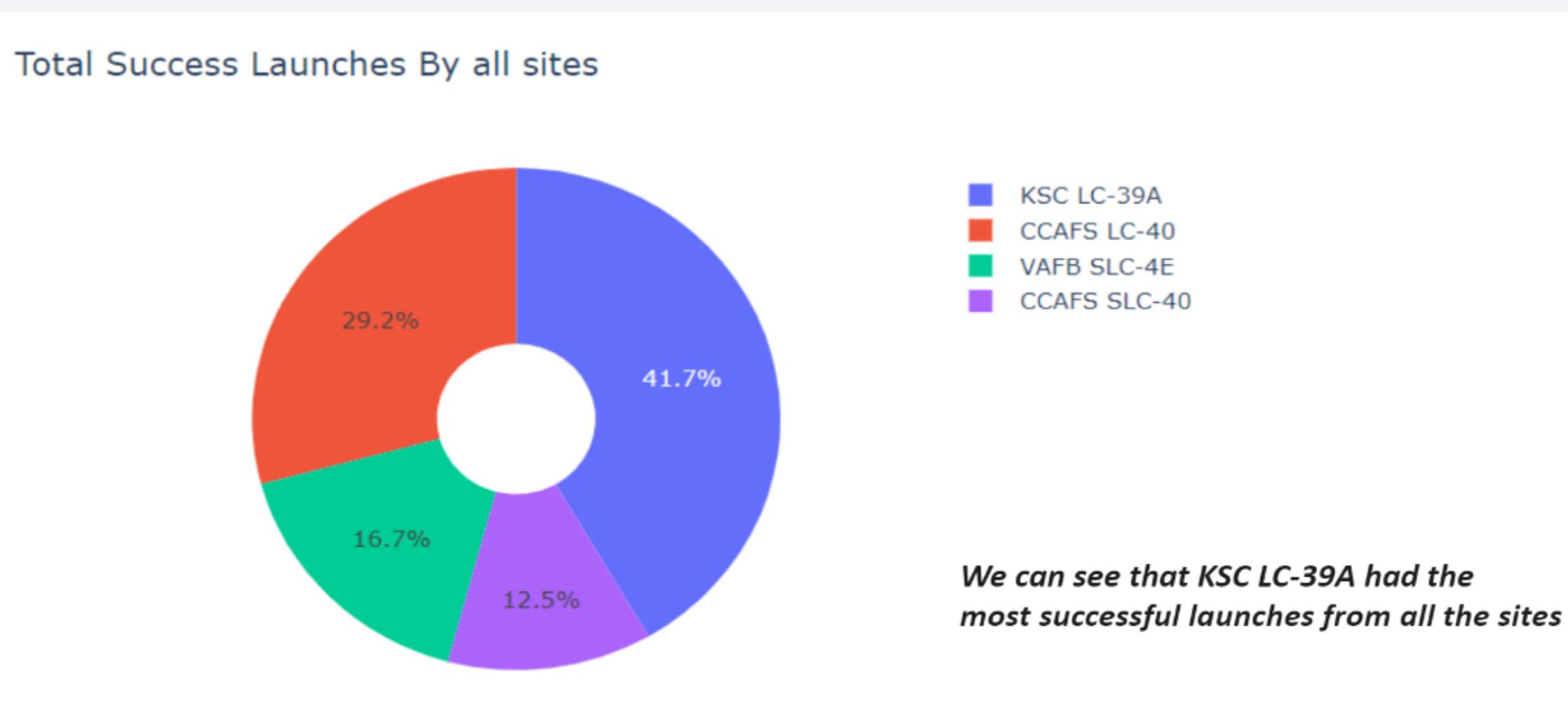


Section 4

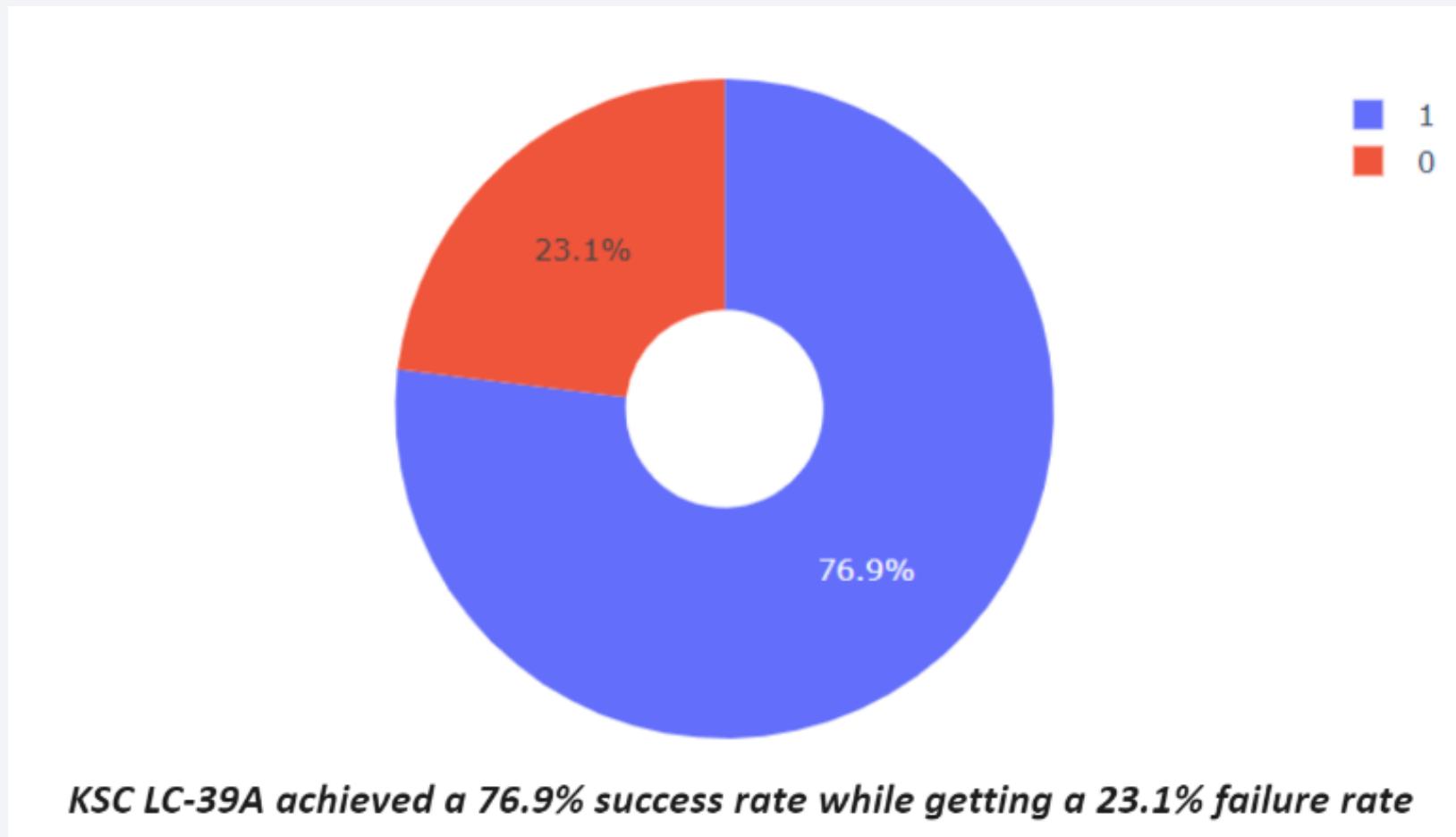
Build a Dashboard with Plotly Dash



Pie chart showing the success percentage achieved by each launch site



Pie chart showing the Launch site with the highest launch success ratio



Scatter plot of Payload vs Class



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with highest classification accuracy.

TASK 8

Create a decision tree classifier object then create a `GridSearchCV` object `tree_cv` with `cv = 10`. Fit the object to find the best parameters from the dictionary `parameters`.

```
In [25]: parameters = {'criterion': ['gini', 'entropy'],
                 'splitter': ['best', 'random'],
                 'max_depth': [2*n for n in range(1,10)],
                 'max_features': ['auto', 'sqrt'],
                 'min_samples_leaf': [1, 2, 4],
                 'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

```
In [26]: # Instantiate the GridSearchCV object: tree_cv
tree_cv = GridSearchCV(tree, parameters, cv=10)

# Fit it to the data
tree_cv.fit(X_train, Y_train)
```

```
In [27]: print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)

tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'best'}
accuracy : 0.875
```

TASK 9

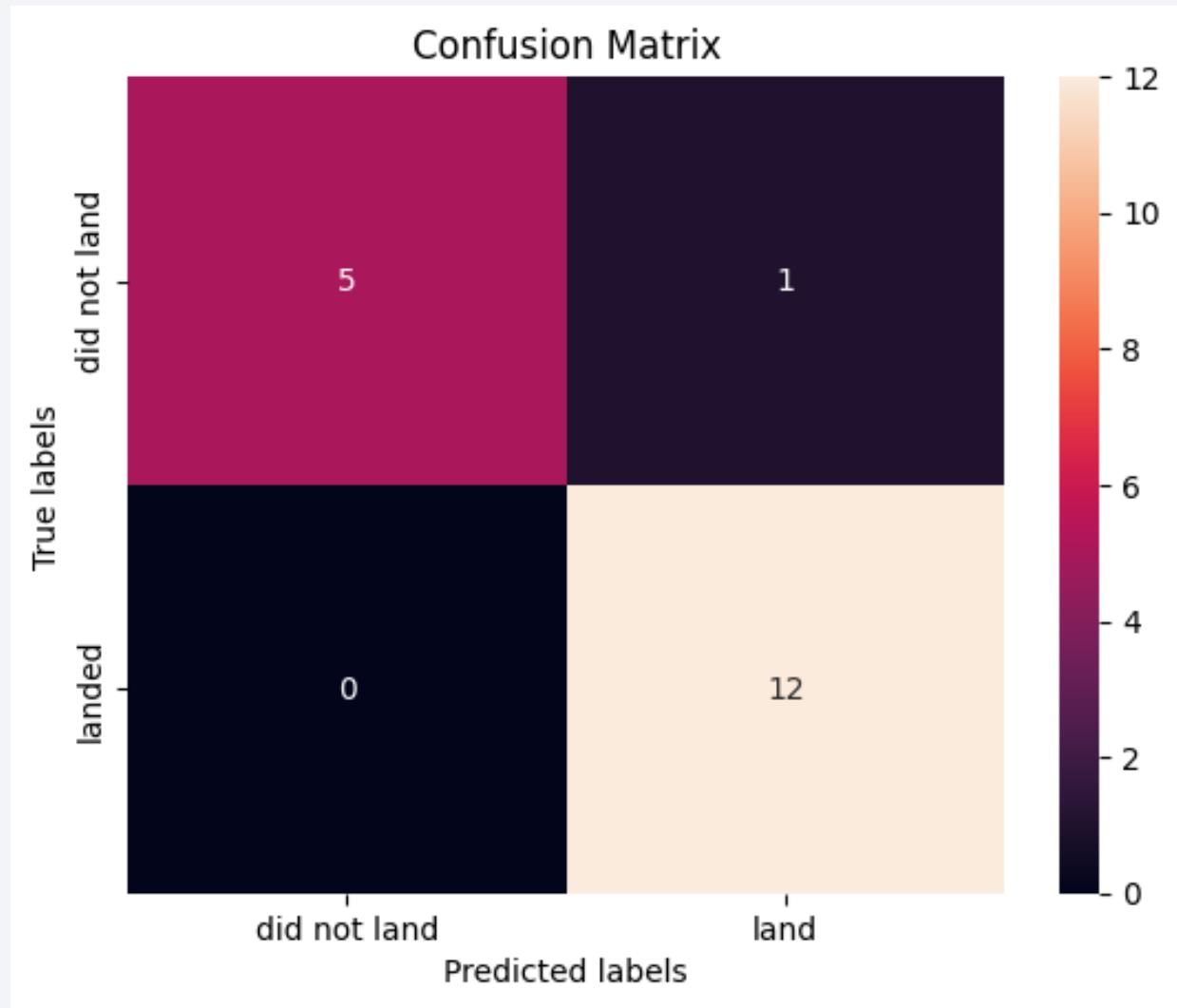
Calculate the accuracy of `tree_cv` on the test data using the method `score`:

```
In [29]: tree_cv.score(X_test,Y_test)
```

```
Out[29]: 0.9444444444444444
```

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!

